Tensor Factorization via Matrix Factorization

Volodymyr Kuleshov*
Arun Tejasvi Chaganty*
Percy Liang

Stanford University

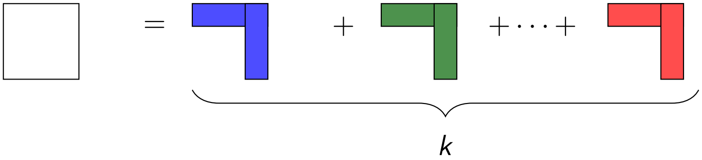
May 8, 2015

What is tensor (CP) factorization?

(Kolda and Bader 2009)

- Tensor analogue of matrix eigen-decomposition.

$$M = \sum_{i=1}^{k} \pi_i u_i \otimes u_i \qquad .$$

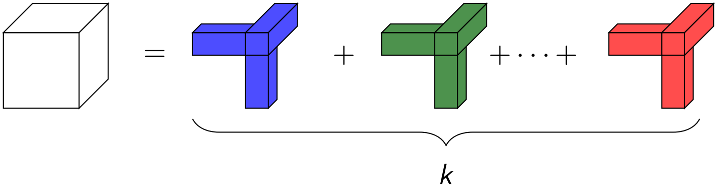


What is tensor (CP) factorization?

(Kolda and Bader 2009)

- Tensor analogue of matrix eigen-decomposition.

$$T = \sum_{i=1}^{k} \pi_i u_i \otimes u_i \otimes u_i \quad .$$



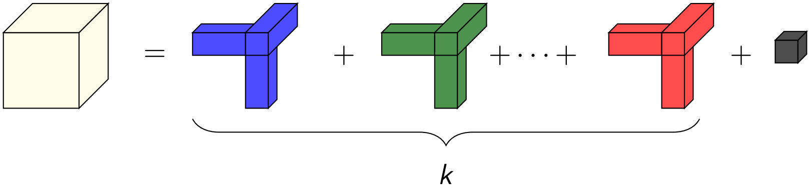
What is tensor (CP) factorization?

(Kolda and Bader 2009)

- Tensor analogue of matrix eigen-decomposition.

$$\widehat{T} = \sum_{i=1}^{k} \pi_i u_i \otimes u_i \otimes u_i + \epsilon R.$$

- **Goal:** Given T with noise, ϵR, **recover factors** u_i.



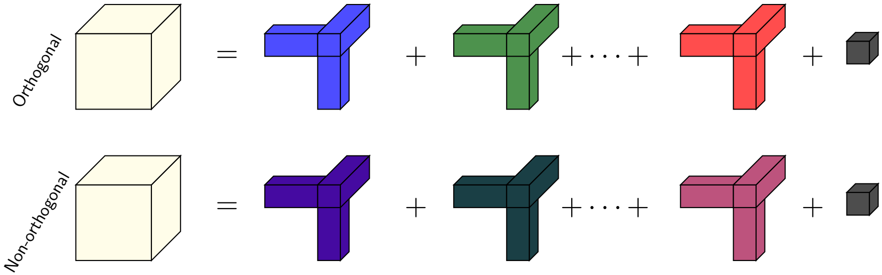
What is tensor (CP) factorization?

(Kolda and Bader 2009)

- Tensor analogue of matrix eigen-decomposition.

$$\widehat{T} = \sum_{i=1}^{k} \pi_i u_i \otimes u_i \otimes u_i + \epsilon R.$$

- **Goal:** Given T with noise, ϵR, **recover factors** u_i.



Why tensor factorization?

- To solve multi-linear algebra problems.

Why tensor factorization?

- To solve multi-linear algebra problems.
- **Parsing**
  - Cohen, Satta, and Collins 2013

Why tensor factorization?

- To solve multi-linear algebra problems.
- **Parsing**
  - Cohen, Satta, and Collins 2013
- **Knowledge base completion**
  - Chang et al. 2014
  - Singh, Rocktäschel, and Riedel 2015

Why tensor factorization?

- To solve multi-linear algebra problems.
- **Parsing**
  - Cohen, Satta, and Collins 2013
- **Knowledge base completion**
  - Chang et al. 2014
  - Singh, Rocktäschel, and Riedel 2015
- **Topic modelling**
  - Anandkumar et al. 2012

Why tensor factorization?

- To solve multi-linear algebra problems.
- **Parsing**
  - Cohen, Satta, and Collins 2013
- **Knowledge base completion**
  - Chang et al. 2014
  - Singh, Rocktäschel, and Riedel 2015
- **Topic modelling**
  - Anandkumar et al. 2012
- **Community detection**
  - Anandkumar et al. 2013a

Why tensor factorization?

- To solve multi-linear algebra problems.
- **Parsing**
  - Cohen, Satta, and Collins 2013
- **Knowledge base completion**
  - Chang et al. 2014
  - Singh, Rocktäschel, and Riedel 2015
- **Topic modelling**
  - Anandkumar et al. 2012
- **Community detection**
  - Anandkumar et al. 2013a
- **Learning latent variable graphical models**
  - Anandkumar et al. 2013b
  - TODO: crowdsourcing
  - TODO: others

Existing tensor factorization algorithms

- **Tensor power method** (Anandkumar et al. 2013b)
  - Analog of matrix power method.
  - Sensitive to noise.
  - Restricted to orthogonal tensors.

Existing tensor factorization algorithms

- **Tensor power method** (Anandkumar et al. 2013b)
  - Analog of matrix power method.
  - Sensitive to noise.
  - Restricted to orthogonal tensors.
- **Alternating least squares** (Comon, Luciani, and Almeida 2009; Anandkumar, Ge, and Janzamin 2014)
  - Sensitive to initialization.

Existing tensor factorization algorithms

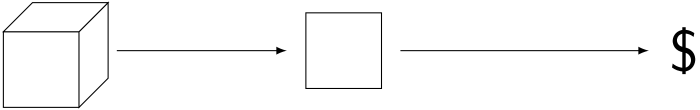
- **Tensor power method** (Anandkumar et al. 2013b)
  - Analog of matrix power method.
  - Sensitive to noise.
  - Restricted to orthogonal tensors.
- **Alternating least squares** (Comon, Luciani, and Almeida 2009; Anandkumar, Ge, and Janzamin 2014)
  - Sensitive to initialization.
- **Both operate on the tensor directly.**

Our approach

- **Objective:** a fast robust algorithm.

Our approach

- **Objective:** a fast robust algorithm.
- **Approach:** use existing fast and robust **matrix** algorithms.



Outline

Tensor factorization

Tensor factorization via matrix factorization

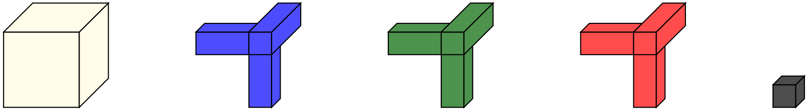
- Single matrix factorizations
- Simultaneous matrix factorizations
- Oracle projections
- Random projections

Non-orthogonal tensor factorization

Empirical results

Conclusions

Tensor factorization via single matrix factorization

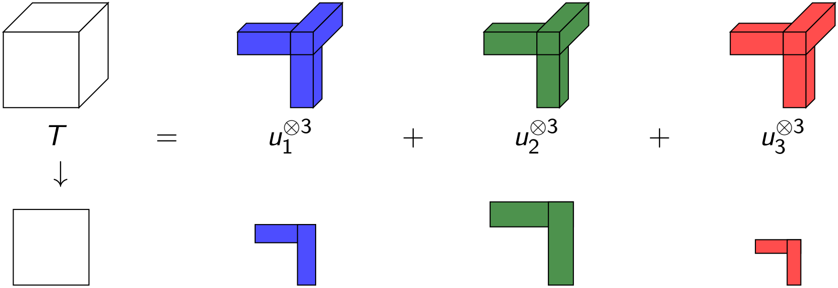


$$T = \pi_1 u_1^{\otimes 3} + \pi_2 u_2^{\otimes 3} + \pi_3 u_3^{\otimes 3} + \epsilon R$$

Tensor factorization via single matrix factorization

$$T = u_1^{\otimes 3} + u_2^{\otimes 3} + u_3^{\otimes 3}$$

Tensor factorization via single matrix factorization

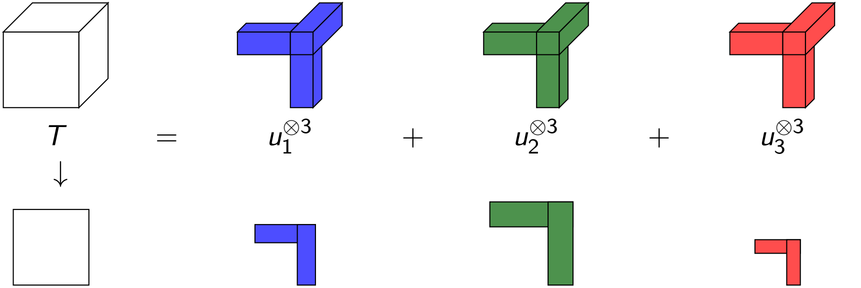


$$T = u_1^{\otimes 3} + u_2^{\otimes 3} + u_3^{\otimes 3}$$

$$\downarrow$$

$$T(I, I, w) = (w^\top u_1) u_1^{\otimes 2} + (w^\top u_2) u_2^{\otimes 2} + (w^\top u_3) u_3^{\otimes 3}$$

Tensor factorization via single matrix factorization



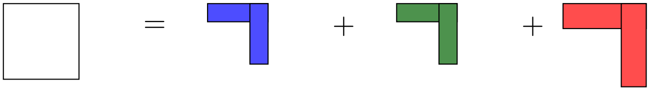
$$T = u_1^{\otimes 3} + u_2^{\otimes 3} + u_3^{\otimes 3}$$

$$\downarrow$$

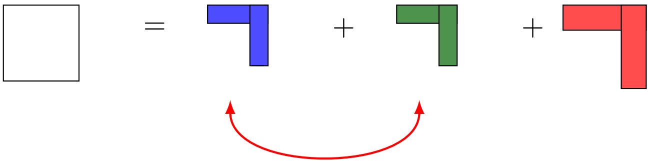
$$T(I, I, w) = \underbrace{(w^\top u_1)}_{\lambda_1} u_1 u_1^\top + \underbrace{(w^\top u_2)}_{\lambda_2} u_2 u_2^\top + \underbrace{(w^\top u_3)}_{\lambda_3} u_3 u_3^\top$$

- **Proposal:** Eigen-decomposition on the projected matrix.

Sensitivity of single matrix projection

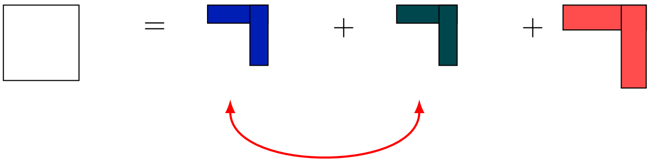


Sensitivity of single matrix projection



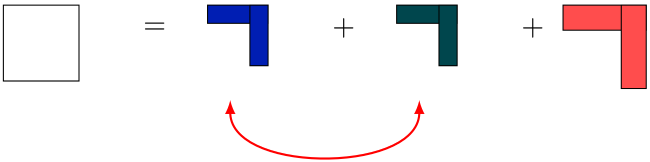
- If two eigenvalues are equal, corresponding eigenvectors are arbitrary.

Sensitivity of single matrix projection



- If two eigenvalues are equal, corresponding eigenvectors are arbitrary.

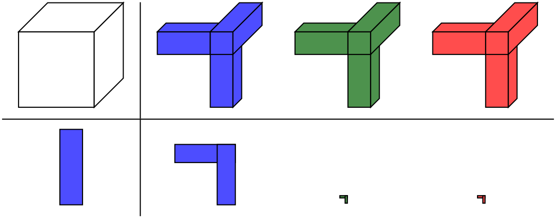
Sensitivity of single matrix projection



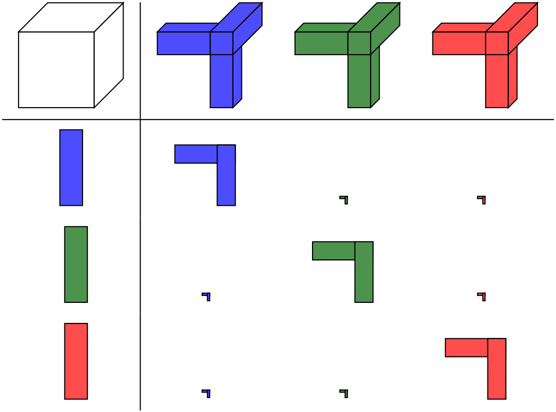
- If two eigenvalues are equal, corresponding eigenvectors are arbitrary.
- **Problem**: Eigendecomposition is very sensitive to the **eigengap**.

$$\text{error in factors} \propto \frac{1}{\min(\text{difference in eigenvalues})}.$$

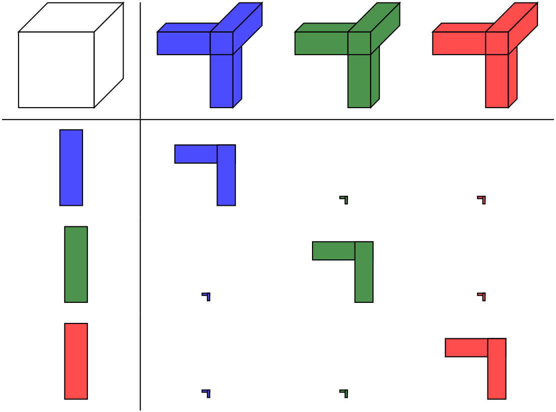
Projections matter



Projections matter



Projections matter



- **How can we leverage multiple projections?**

Outline

Tensor factorization

Tensor factorization via matrix factorization
- Single matrix factorizations
- Simultaneous matrix factorizations
- Oracle projections
- Random projections

Non-orthogonal tensor factorization

Empirical results

Conclusions

Enter simultaneous diagonalization

$$\underbrace{T(I, I, w_1)}_{M_1} = \underbrace{(w_1^\top u_1)}_{\lambda_{11}} u_1 u_1^\top + \underbrace{(w_1^\top u_2)}_{\lambda_{21}} u_2 u_2^\top + \underbrace{(w_1^\top u_3)}_{\lambda_{31}} u_3 u_3^\top$$

Enter simultaneous diagonalization

$$\underbrace{T(I, I, w_1)}_{M_1} = \underbrace{(w_1^\top u_1)}_{\lambda_{11}} u_1 u_1^\top + \underbrace{(w_1^\top u_2)}_{\lambda_{21}} u_2 u_2^\top + \underbrace{(w_1^\top u_3)}_{\lambda_{31}} u_3 u_3^\top$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\underbrace{T(I, I, w_I)}_{M_I} = \underbrace{(w_I^\top u_1)}_{\lambda_{11}} u_1 u_1^\top + \underbrace{(w_I^\top u_2)}_{\lambda_{21}} u_2 u_2^\top + \underbrace{(w_I^\top u_3)}_{\lambda_{31}} u_3 u_3^\top$$

Enter simultaneous diagonalization

$$\underbrace{T(I, I, w_1)}_{M_1} = \underbrace{(w_1^\top u_1)}_{\lambda_{11}} u_1 u_1^\top + \underbrace{(w_1^\top u_2)}_{\lambda_{21}} u_2 u_2^\top + \underbrace{(w_1^\top u_3)}_{\lambda_{31}} u_3 u_3^\top$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\underbrace{T(I, I, w_I)}_{M_I} = \underbrace{(w_I^\top u_1)}_{\lambda_{11}} u_1 u_1^\top + \underbrace{(w_I^\top u_2)}_{\lambda_{21}} u_2 u_2^\top + \underbrace{(w_I^\top u_3)}_{\lambda_{31}} u_3 u_3^\top$$

- **Projections share factors.**

Algorithm

- **Algorithm:** Simultaneously diagonalize projected matrices.

$$\widehat{U} = \arg\max_{\widehat{U}} \sum_{l=1}^{L} \text{off}(U^\top M_l U) \qquad \text{off}(A) = \sum_{i \neq j} A_{ij}^2.$$

Algorithm

- **Algorithm:** Simultaneously diagonalize projected matrices.

$$\widehat{U} = \arg\max_{\widehat{U}} \sum_{l=1}^{L} \mathrm{off}(U^\top M_l U) \qquad \mathrm{off}(A) = \sum_{i \neq j} A_{ij}^2.$$

- Optimize using the Jacobi angles (Cardoso and Souloumiac 1996).

Algorithm

- **Algorithm:** Simultaneously diagonalize projected matrices.

$$\widehat{U} = \arg\max_{\widehat{U}} \sum_{l=1}^{L} \mathrm{off}(U^\top M_l U) \qquad \mathrm{off}(A) = \sum_{i \neq j} A_{ij}^2.$$

- Optimize using the Jacobi angles (Cardoso and Souloumiac 1996).
- Multiple projections proposed in Anandkumar, Hsu, and Kakade 2012, but didn't use simultaneous diagonalization.

Comparison with single matrix factorization

- Single matrix factorization depends on **minimum eigengap**.

$$\text{error in factors} \propto \frac{1}{\min_{i,j} \text{difference in eigenvalues}}.$$

Comparison with single matrix factorization

- Single matrix factorization depends on **minimum eigengap**.

$$\text{error in factors} \propto \frac{1}{\min_{i,j} \text{difference in eigenvalues}}.$$

- Simultaneous matrix factorization depends on **average eigengap**.

$$\text{error in factors} \propto \frac{1}{\min_{i,j} \text{average difference in eigenvalues}}.$$

Comparison with single matrix factorization

- Single matrix factorization depends on **minimum eigengap**.

$$\text{error in factors} \propto \frac{1}{\min_{i,j} \quad |\lambda_i - \lambda_j|}.$$

- Simultaneous matrix factorization depends on **average eigengap**.

$$\text{error in factors} \propto \frac{1}{\min_{i,j} \quad \sum_{l=1}^{L} |\lambda_{il} - \lambda_{jl}|}.$$

Outline

Tensor factorization

Tensor factorization via matrix factorization

- Single matrix factorizations
- Simultaneous matrix factorizations
- Oracle projections
- Random projections

Non-orthogonal tensor factorization

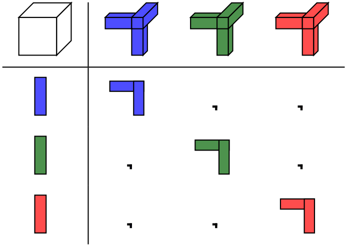
Empirical results

Conclusions

Oracle projections

Theorem

Pick k projections along the factors (u_i). Then,

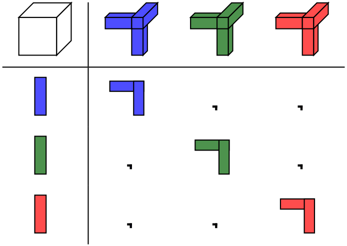


Oracle projections

Theorem

Pick k projections along the factors (u_i). Then,

$$\textit{error in factors} \leq O\left(\frac{\sqrt{\pi_{\max}}}{\pi_{\min}^2}\right)\epsilon.$$

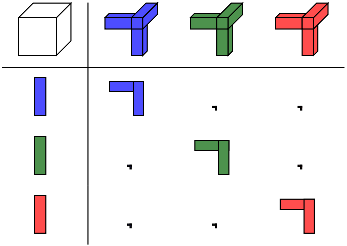


Oracle projections

Theorem

Pick k projections along the factors (u_i). Then,

$$\text{error in factors} \leq O\left(\frac{\sqrt{\pi_{\max}}}{\pi_{\min}^2}\right)\epsilon.$$



Outline

Tensor factorization

Tensor factorization via matrix factorization

- Single matrix factorizations
- Simultaneous matrix factorizations
- Oracle projections
- Random projections

Non-orthogonal tensor factorization

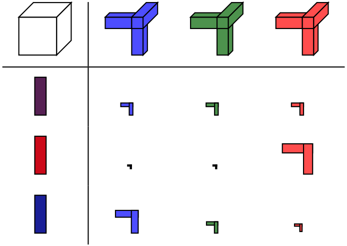
Empirical results

Conclusions

Random projections

Theorem

Pick $O(k \log k)$ projections **randomly from the unit sphere**. *Then, with probability $> 1 - \delta$,*

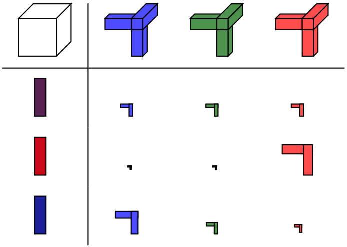


Random projections

Theorem

Pick $O(k \log k)$ projections **randomly from the unit sphere**. *Then, with probability $> 1 - \delta$,*

$$\text{error in factors} \leq O\left(\frac{\sqrt{\pi_{\max}}}{\pi_{\min}^2}\right)\epsilon + C(\delta)\epsilon$$

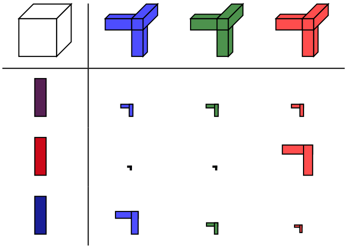


Random projections

Theorem

Pick $O(k \log k)$ projections **randomly from the unit sphere**. *Then, with probability $> 1 - \delta$,*

$$\text{error in factors} \leq O\left(\frac{\sqrt{\pi_{\max}}}{\pi_{\min}^2}\right)\epsilon + C(\delta)\epsilon$$

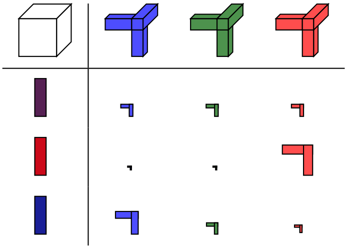


Random projections

Theorem

Pick $O(k \log k)$ *projections* **randomly from the unit sphere**. *Then, with probability* $> 1 - \delta$,

$$\textit{error in factors} \leq O\left(\frac{\sqrt{\pi_{\max}}}{\pi_{\min}^2}\right)\epsilon + C(\delta)\epsilon$$



- As good as having oracle projections!

Final algorithm

- **Algorithm**:

Final algorithm

- **Algorithm**:
  - Project tensor on to $O(k \log k)$ random vectors.

Final algorithm

- **Algorithm**:
  - Project tensor on to $O(k \log k)$ random vectors.
  - Recover approximate factors $\tilde{u}_i^{(0)}$ through simultaneous diagonalization.

Final algorithm

- **Algorithm**:
  - Project tensor on to $O(k \log k)$ random vectors.
  - Recover approximate factors $\tilde{u}_i^{(0)}$ through simultaneous diagonalization.
  - Project tensor on to approximated factors.

Final algorithm

- **Algorithm**:
  - Project tensor on to $O(k \log k)$ random vectors.
  - Recover approximate factors $\tilde{u}_i^{(0)}$ through simultaneous diagonalization.
  - Project tensor on to approximated factors.
  - Return factors $\tilde{u}_i$ from simultaneous diagonalization.

Outline

Tensor factorization

Tensor factorization via matrix factorization

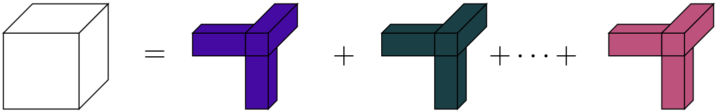
- Single matrix factorizations
- Simultaneous matrix factorizations
- Oracle projections
- Random projections

Non-orthogonal tensor factorization

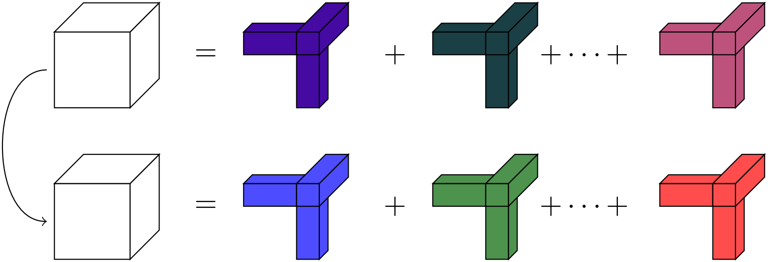
Empirical results

Conclusions

Naive approach: whitening non-orthogonal factors

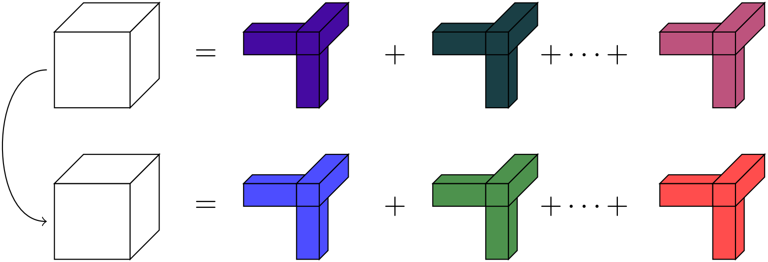


Naive approach: whitening non-orthogonal factors



- Use a whitening transformation to orthogonalize tensor (Anandkumar et al. 2013b).

Naive approach: whitening non-orthogonal factors



- Use a whitening transformation to orthogonalize tensor (Anandkumar et al. 2013b).
  - Is a major source of errors itself (Souloumiac 2009).

Non-orthogonal simultaneous diagonalization

$$\underbrace{T(I, I, w_1)}_{M_1} = \underbrace{(w_1^\top u_1)}_{\lambda_{11}} u_1 u_1^\top + \underbrace{(w_1^\top u_2)}_{\lambda_{21}} u_2 u_2^\top + \underbrace{(w_1^\top u_3)}_{\lambda_{31}} u_3 u_3^\top$$

- No unique non-orthogonal factorization for a single matrix.

Non-orthogonal simultaneous diagonalization

$$\underbrace{T(I, I, w_1)}_{M_1} = \underbrace{(w_1^\top u_1)}_{\lambda_{11}} u_1 u_1^\top + \underbrace{(w_1^\top u_2)}_{\lambda_{21}} u_2 u_2^\top + \underbrace{(w_1^\top u_3)}_{\lambda_{31}} u_3 u_3^\top$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\underbrace{T(I, I, w_I)}_{M_I} = \underbrace{(w_I^\top u_1)}_{\lambda_{1I}} u_1 u_1^\top + \underbrace{(w_I^\top u_2)}_{\lambda_{2I}} u_2 u_2^\top + \underbrace{(w_I^\top u_3)}_{\lambda_{3I}} u_3 u_3^\top$$

- No unique non-orthogonal factorization for a single matrix.
- ≥ 2 **matrices have a unique non-orthogonal factorization.**

Non-orthogonal simultaneous diagonalization

$$\underbrace{T(I,I,w_1)}_{M_1} = \underbrace{(w_1^\top u_1)}_{\lambda_{11}} u_1 u_1^\top + \underbrace{(w_1^\top u_2)}_{\lambda_{21}} u_2 u_2^\top + \underbrace{(w_1^\top u_3)}_{\lambda_{31}} u_3 u_3^\top$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\underbrace{T(I,I,w_I)}_{M_I} = \underbrace{(w_I^\top u_1)}_{\lambda_{1I}} u_1 u_1^\top + \underbrace{(w_I^\top u_2)}_{\lambda_{2I}} u_2 u_2^\top + \underbrace{(w_I^\top u_3)}_{\lambda_{3I}} u_3 u_3^\top$$

- No unique non-orthogonal factorization for a single matrix.
- ≥ 2 **matrices have a unique non-orthogonal factorization.**
- **Note:** λ_{iI} are factor weights, not eigenvalues.

Non-orthogonal simultaneous diagonalization

- **Algorithm:** Simultaneously diagonalize projected matrices.

$$\widehat{U} = \arg\max_{\widehat{U}} \sum_{l=1}^{L} \mathrm{off}(U^{-1} M_l U^{-\top}) \qquad \mathrm{off}(A) = \sum_{i \neq j} A_{ij}^2.$$

Non-orthogonal simultaneous diagonalization

- **Algorithm:** Simultaneously diagonalize projected matrices.

$$\widehat{U} = \arg\max_{\widehat{U}} \sum_{l=1}^{L} \mathrm{off}(U^{-1} M_l U^{-\top}) \qquad \mathrm{off}(A) = \sum_{i \neq j} A_{ij}^2.$$

- U are not constrained to be orthogonal.

Non-orthogonal simultaneous diagonalization

- **Algorithm:** Simultaneously diagonalize projected matrices.

$$\widehat{U} = \arg\max_{\widehat{U}} \sum_{l=1}^{L} \mathrm{off}(U^{-1} M_l U^{-\top}) \qquad \mathrm{off}(A) = \sum_{i \neq j} A_{ij}^2.$$

- U are not constrained to be orthogonal.
- Optimize using the QR1JD algorithm (Souloumiac 2009).
  - Only guaranteed to have local convergence.

Results: Non-orthogonal simultaneous diagonalization

Theorem (Oracle projections)

Pick k projections along the factors (u_i). Then,

$$\textit{error in factors} \leq O\left(\|U^{-\top}\|_2^3 \frac{\sqrt{\pi_{\max}}}{\pi_{\min}^2} \right) \epsilon,$$

where $U = [u_1 | \cdots | u_k]$.

Results: Non-orthogonal simultaneous diagonalization

Theorem (Oracle projections)

Pick k projections along the factors (u_i). Then,

$$\textit{error in factors} \leq O\left(\|U^{-\top}\|_2^3 \frac{\sqrt{\pi_{\max}}}{\pi_{\min}^2}\right)\epsilon,$$

where $U = [u_1|\cdots|u_k]$.

Outline

Tensor factorization

Tensor factorization via matrix factorization
- Single matrix factorizations
- Simultaneous matrix factorizations
- Oracle projections
- Random projections

Non-orthogonal tensor factorization

Empirical results

Conclusions

Outline

Tensor factorization

Tensor factorization via matrix factorization

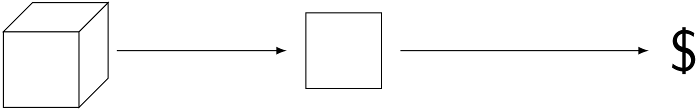
- Single matrix factorizations
- Simultaneous matrix factorizations
- Oracle projections
- Random projections

Non-orthogonal tensor factorization

Empirical results

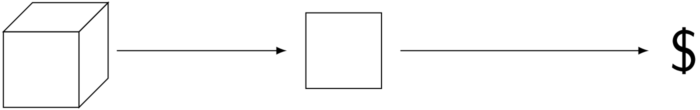
Conclusions

Conclusions



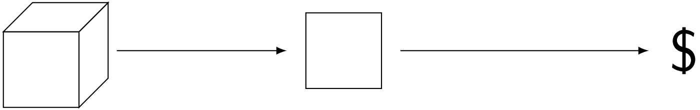
- Reduce tensor problems to matrix ones with $\tilde{O}(k)$ random projections.

Conclusions



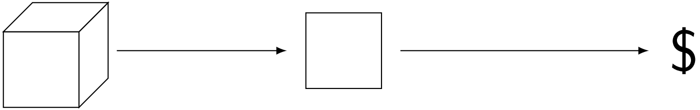
- Reduce tensor problems to matrix ones with $\tilde{O}(k)$ random projections.
- **Robust** to noise with general support for non-orthogonal factors.

Conclusions



- Reduce tensor problems to matrix ones with $\tilde{O}(k)$ random projections.
- **Robust** to noise with general support for non-orthogonal factors.
- **Competitive empirical** performance.

Conclusions



- Reduce tensor problems to matrix ones with $\tilde{O}(k)$ random projections.
- **Robust** to noise with general support for non-orthogonal factors.
- **Competitive empirical** performance.
- **Questions?**